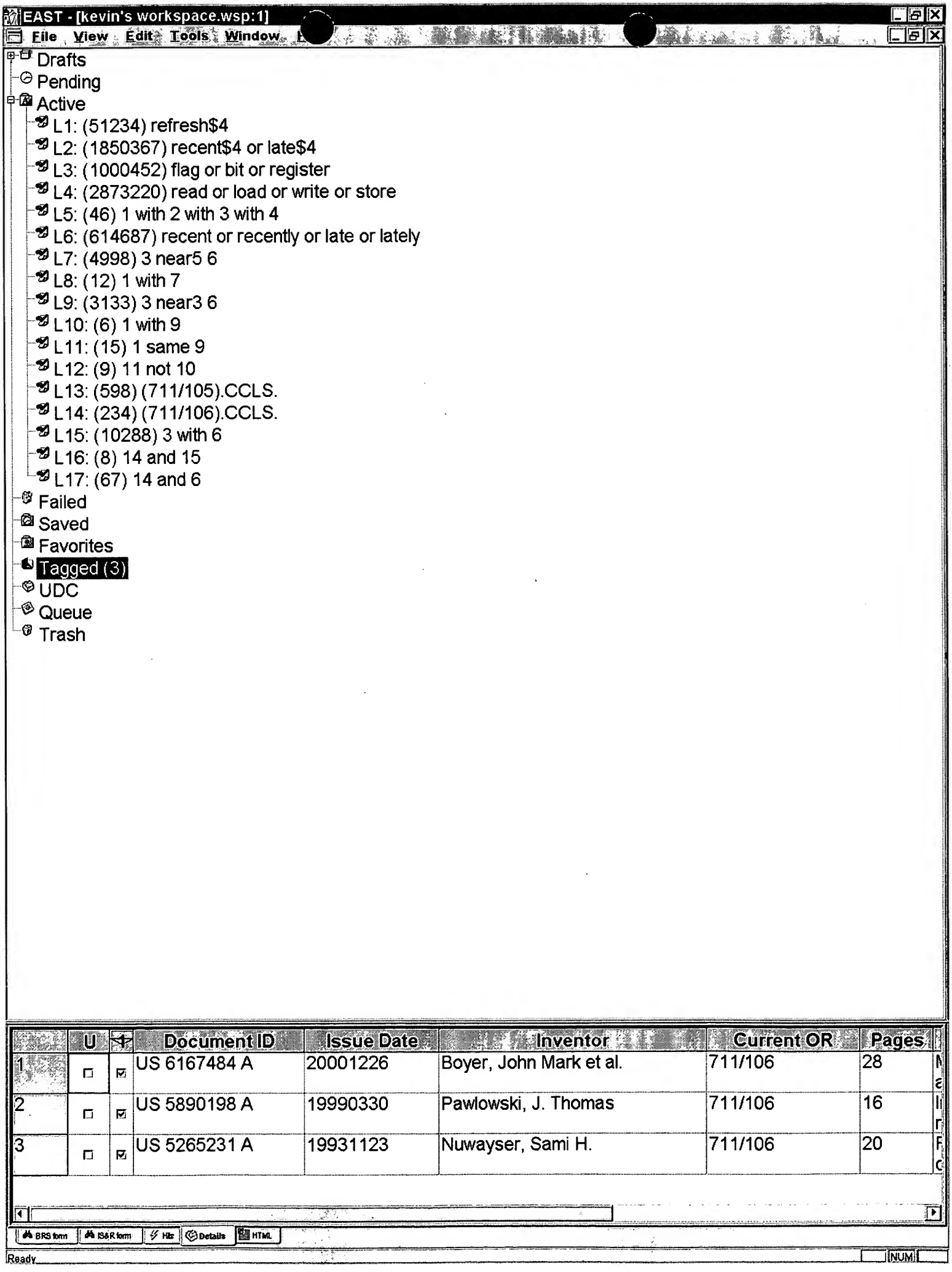
L Number	Hits	Search Text	DB	Time stamp
1	51234	refresh$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 11:21
2	1850367	recent$4 or late$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 11:47
3	1000452	flag or bit or register	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 11:31
4	2873220	read or load or write or store	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 11:31
5	46	refresh$4 with (recent$4 or late$4) with (flag or bit or register) with (read or load or write or store)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 11:33
6	614687	recent or recently or late or lately	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 11:46
7	4998	(flag or bit or register) near5 (recent or recently or late or lately)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 11:51
8	12	refresh$4 with ((flag or bit or register) near5 (recent or recently or late or lately))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 11:47
9	3133	(flag or bit or register) near3 (recent or recently or late or lately)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 11:51
10	6	refresh$4 with ((flag or bit or register) near3 (recent or recently or late or lately))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 13:18
11	15	refresh$4 same ((flag or bit or register) near3 (recent or recently or late or lately))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 13:18
12	9	(refresh$4 same ((flag or bit or register) near3 (recent or recently or late or lately))) not (refresh$4 with ((flag or bit or register) near3 (recent or recently or late or lately)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 13:21
13	598	(711/105).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 13:20

14	234	(711/106).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 13:20
15	10288	(flag or bit or register) with (recent or recently or late or lately)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 13:21
16	8	((711/106).CCLS.) and ((flag or bit or register) with (recent or recently or late or lately))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 13:26
17	67	((711/106).CCLS.) and (recent or recently or late or lately)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/13 13:27


EAST - [kevin's workspace.wsp:1]
File View Edit Tools Window
Drafts
Pending
Active
L1: (51234) refresh$4
L2: (1850367) recent$4 or late$4
L3: (1000452) flag or bit or register
L4: (2873220) read or load or write or store
L5: (46) 1 with 2 with 3 with 4
L6: (614687) recent or recently or late or lately
L7: (4998) 3 near5 6
L8: (12) 1 with 7
L9: (3133) 3 near3 6
L10: (6) 1 with 9
L11: (15) 1 same 9
L12: (9) 11 not 10
L13: (598) (711/105).CCLS.
L14: (234) (711/106).CCLS.
L15: (10288) 3 with 6
L16: (8) 14 and 15
L17: (67) 14 and 6
Failed
Saved
Favorites
Tagged (3)
UDC
Queue
Trash
U
Document ID
Issue Date
Inventor
Current OR
Pages
1 US 6167484 A 20001226 Boyer, John Mark et al. 711/106 28
2 US 5890198 A 19990330 Pawlowski, J. Thomas 711/106 16
3 US 5265231 A 19931123 Nuwayser, Sami H. 711/106 20
BRS form
IS&R form
Hits
Details
HTML
Ready
NUM

US-PAT-NO: 5890198

DOCUMENT-IDENTIFIER: US 5890198 A
See image for Certificate of Correction

TITLE: Intelligent refresh controller for dynamic memory devices

--------- KWIC --------

Brief Summary Text - BSTX (9):
More recently, DRAMs have been proposed which have a self-refresh mode for executing the refresh process. In such DRAMs, a refresh signal does not need to be generated by the memory controller or system processor. DRAMs having self-refresh capability include a circuit specifically designed for initiating and carrying out the refresh operation.

Brief Summary Text - BSTX (15):
The refresh data cells are updated whenever the row is accessed, either as initiated by the refresh controller or when an external request is executed. The net result is that refresh only occurs when necessary. If the row has been accessed for either a read or a write operation, the "count down" to the next required refresh operation begins anew. Unnecessary refreshes are avoided because rows which have been recently accessed are not refreshed.

Current US Original Classification - CCOR (1):
711/106

US-PAT-NO: 5265231

DOCUMENT-IDENTIFIER: US 5265231 A

TITLE: Refresh control arrangement and a method for refreshing a plurality of random access memory banks in a memory system

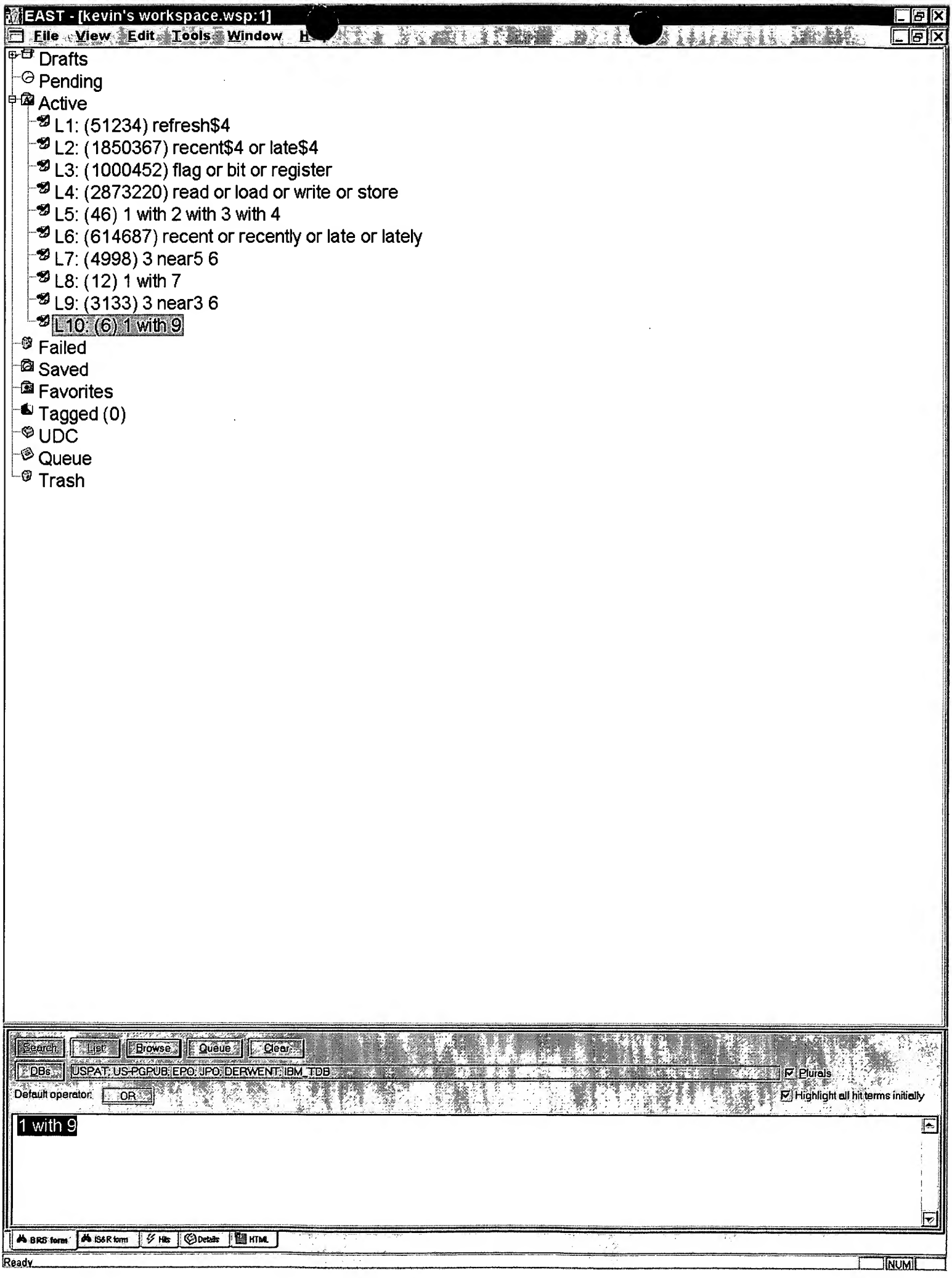
--------- KWIC --------

Brief Summary Text - BSTX (8):

More recently, the DRAM chips have been developed which include the refresh counters. For those chips, refresh is enabled, by external control circuitry, by providing the RAS signal and the CAS signal in reverse order, that is, by providing the CAS signal a selected time before the RAS signal. The DRAM chips, upon receiving the CAS and RAS signals in that manner, refresh the particular rows identified by their respective refresh counters.

Current US Original Classification - CCOR (1):

711/106


EAST - [kevin's workspace.wsp:1]
File View Edit Tools Window H
Drafts
Pending
Active
L1: (51234) refresh$4
L2: (1850367) recent$4 or late$4
L3: (1000452) flag or bit or register
L4: (2873220) read or load or write or store
L5: (46) 1 with 2 with 3 with 4
L6: (614687) recent or recently or late or lately
L7: (4998) 3 near5 6
L8: (12) 1 with 7
L9: (3133) 3 near3 6
L10: (6) 1 with 9
Failed
Saved
Favorites
Tagged (0)
UDC
Queue
Trash
Search List Browse Queue Clear
DBs USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
Plurals
Default operator: OR
Highlight all hit terms initially
1 with 9
BRS form IS&R form Hits Details HTML
Ready
NUM

US 20030005226A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0005226 A1**

Hong (43) **Pub. Date: Jan. 2, 2003**

(54) **MEMORY MANAGEMENT APPARATUS AND METHOD**

(75) Inventor: **Joseph N. Hong**, Chandler, AZ (US)

Correspondence Address:
Schwegman, Lundberg, Woessner & Kluth, P.A.
P.O. Box 2938
Minneapolis, MN 55402 (US)

(73) Assignee: **Intel Corporation**

(21) Appl. No.: **09/896,597**

(22) Filed: **Jun. 29, 2001**

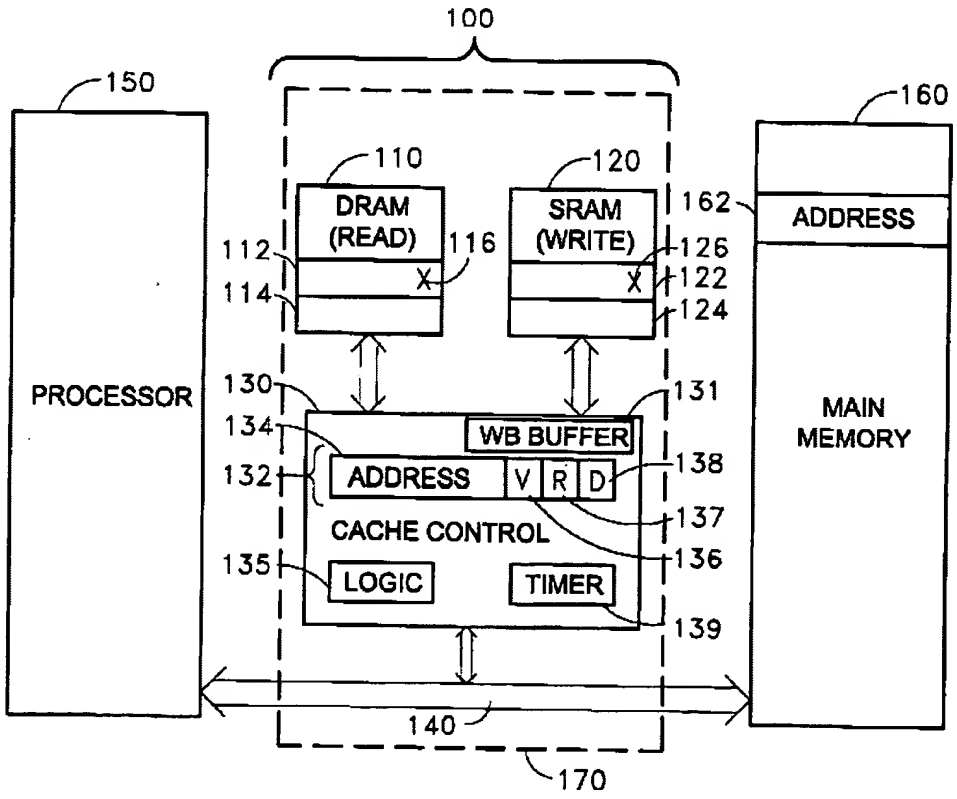
Publication Classification

(51) **Int. Cl.**[7] **G06F 13/00**

(52) **U.S. Cl.** **711/119**; 711/133

(57) **ABSTRACT**

Cache memory systems, microprocessors, and computer systems, as well as methods of operating a cache memory system are described. The cache memory system includes a cache memory controller coupled to a dynamic cache memory and a static cache memory. The cache memory system provides the advantages of using dynamic memory (having a small circuit real estate requirement) for cache read operations, and static memory for cache write operations. Using the static memory for cache write operations allows the cache memory system to function as a write-back cache, instead of an instruction-only, or write-through cache.



DOCUMENT-IDENTIFIER: US 20030005226 A1

TITLE: Memory management apparatus and method

-------- KWIC -------

Detail Description Paragraph - DETX (23):

[0034] Thus, with regard to the first table 200, the VALID bits 232, 234, 236, and 238 reflect the status of the operands contained within the memory locations of the dynamic cache memory corresponding to the main memory addresses referenced by ADD0, ADD1, ADD2, and ADD3, respectively. That is, the operands contained in the dynamic cache memory locations corresponding to ADD0, ADD2, and ADD3 are invalid, while the operand contained in the dynamic cache memory location corresponding to ADD1 is valid. Assuming that the entire dynamic cache memory has been accessed (read from or written to) recently, all of the REFRESH bits 222, 224, 226, and 228 are set. As can be seen in table 210, however, some time after the refresh operation has occurred, but before a new refresh operation is conducted, REFRESH bits 222, 224, and 228 are reset, indicating additional accesses (i.e., either read or write operations) to the dynamic cache memory locations corresponding to ADD0, ADD1, and ADD3. Thus, as can be seen in table 220, just before the end of the refresh interval, the VALID bit 236 will be reset to indicate that the content of the dynamic cache memory location corresponding to ADD2 is invalid (i.e., the content of the dynamic cache memory corresponding to ADD2 has now been selectively invalidated), because there has been no read or write access to this location since the beginning of the current refresh interval.

US-PAT-NO: 6167484

DOCUMENT-IDENTIFIER: US 6167484 A

TITLE: Method and apparatus for leveraging history bits to optimize memory refresh performance

---------- KWIC ---------

Detailed Description Text - DETX (3):

Generally, the present invention is a method and apparatus for reducing the power consumed by DRAM refresh operations and/or improving the system access (i.e., read/write) operational bandwidth of an embedded DRAM. These advantages are obtained by retaining history information for each memory row or refresh group of memory cells within the embedded DRAM array. When time comes to refresh a refresh group or a memory row, this history information is processed whereby refreshing is not performed on a row of cells that do not need refreshing (i.e., rows that are inactive, or rows that were **recently** read or written since these accesses inherently do a refresh of the rows). In one embodiment, time that is not used refreshing, for whatever reason, is given back to the CPU/system to improve system access bandwidth to the DRAM. In another embodiment, the reduced number of refreshes will reduce power consumed by the embedded memory which is an advantage in any microcontroller solution. A significant portion of the roughly 10% refresh bandwidth consumed by embedded DRAM may be returned to the system by the algorithms and systems taught herein.

Detailed Description Text - DETX (16):

A system access to the row associated with the specific history bit i will result in both the SYS.sub.-- ACCESS and DROW.sub.i signals for that one specific history bit i being asserted. When the DROW.sub.i and SYS.sub.-- ACCESS signals are asserted, the AND gate 306 generates a 1 which asynchronously sets the flip-flop 302 (Q=1 and QB=0) indicating that the row has been **recently** refreshed by virtue of a system access (read or write). If the DRAM row associated with the cell 204a is refreshed in due course, then REF and DROW.sub.i will be asserted for that history bit i. When REF and DROW.sub.i are both asserted and DROW.sub.i is subsequently negated, the NAND gate 304 output is a negative leading edge pulse which causes flip-flop 302 to toggle to the opposite state on the positive trailing edge of the pulse. Therefore, if the prior state of flip-flop 302 was 0 (i.e., stale) it would be toggled to 1 (i.e., fresh) indicating that it need not be refreshed in the next MROC. If flip-flop 302 was initially a 1 (i.e., fresh), it would be toggled to 0 (i.e., stale) which schedules the row to be refreshed in the next MROC. Because the flip-flop 302 is toggled on the trailing edge of the pulse, the flip-flop changes state only after an operation has been completed.

Detailed Description Text - DETX (52):

History state 614 illustrates the effect of a refresh not needing to occur during a specific MROC due to a read/write operation in time region 20 of FIG.

5. Specifically, since history bit 1 at the beginning of time 12c-1 was set by a previous read or write, a refresh that would otherwise have occurred during time 12c-1 is canceled, and history bit 1 is cleared as illustrated in state 613 to indicate that a next MROC should refresh the row 1 (unless another read or write subsequently occurs). By not refreshing the row 1 associated with history bit 1 immediately following an access to the row, power savings over the prior art is realized. In other words, by keeping track of read and write history to a DRAM row, refresh operations on these rows may be delayed thereby saving power. The power savings is realized because the memory cells associated with the **recently** written or read memory row do not need to be refreshed immediately following the access. A read or write access in the DRAM architecture inherently performs a refresh operation.

Detailed Description Text - DETX (54):

State 617 of FIG. 7 shows the logical values of the history bits at the end of the MROC c of FIG. 5. Due to the reads and writes, in time periods 20 and 22 of FIG. 5 that created the bracketed `1` values in states 607 and 609, the rows 1, 2, 4, 6, and 7 needed no refresh in MROC c where refreshing is deferred to MROC d for these five memory rows. Therefore, power can be saved by delaying or preempting the refresh of rows that have been **recently** refreshed by a read or write operation.

Detailed Description Text - DETX (89):

Therefore, it should be evident that the present invention provides an advantage over the prior art in that it allows for increased system access bandwidth for reads and writes by delaying or avoiding unneeded refresh operations onto already fresh memory rows. The increase in system access bandwidth is realized by refreshing, within a plurality of cluster refresh opportunity times (CROTs), only those rows actually in need of refreshing. Refresh time is only allocated to those memory rows or refresh groups of DRAM cells that actually need a refresh operation. This is unlike the prior which actually dedicates time to refresh each row at a specific time whether or not the refresh is needed. In other words, the prior art does not keep track of whether a **recent** read or write can be used to cancel an upcoming refresh of a memory row. In addition, this embodiment of FIGS. 9-13 is an improvement over the first embodiments disclosed in FIGS. 1-8 when additional system access bus bandwidth is desired.

Current US Original Classification - CCOR (1):

711/106